

REMARKS

This Amendment is submitted in response to the Examiner's Action dated April 20, 2001, having a shortened statutory period set to expire July 20, 2001.

In the action dated April 20, 2001, the Examiner notes the reference to related applications present at the first sentence of the Specification. This reference is to related applications which Applicant includes without claiming priority from those applications. The serial number and filing date of each application has been included by this Amendment.

Next, the Examiner has rejected Claims 1-3, 12-13 and 19-20 under 35 U.S.C. § 102(e) as being anticipated by *Li, et al.*, U.S. Patent No. 6,205,418. That rejection is respectfully traversed.

The invention in the present application is directed to a novel "text string data structure" which includes multiple fields wherein the content of each field is specified as set forth within the claims. A "text string" is a term of art in the computer technology area and as evidence of this fact, Applicant submits herewith page 451 of the Microsoft Press Computer Dictionary, Third Edition, which defines "string" as "a data structure composed of a sequence in characters usually representing human-readable text." Further evidence of the uniqueness of this verbiage is set forth within *Li, et al.*, at column 2, lines 23, *et seq.*, wherein *Li, et al.*, state "as an example, a PC-based attendant console application includes many embedded text strings that could not be intercepted by these types of multiple language systems." (emphasis added) It is precisely this problem which is addressed by the present Application in that translation system such as those disclosed within *Li, et al.*, are well known; however, such systems require a separate translation database to transform imbedded text strings found within an application or operating system. It is an object of the present Application to obviate that necessity by providing a novel, useful and

unobvious "text string data structure" which includes multiple fields where each field includes a different content, as set forth within the present claims.

Specifically, with respect to Claim 1, the multi-field text string data object is expressly recited as including a first field "containing a first character string representing a word" and a second field "containing a second character string representing the word." In support of the rejection, the Examiner cites Table 2 of *Li, et al.*, which illustrates, in graphic form, the transform database utilized by the translation program of *Li, et al.* This transform database contains three fields, one field of which identifies a particular language. A second field of which contains a particular translation code which corresponds to an existing text string and the third field corresponds to a translation of that existing text string into the language identified by the language code. Applicant respectfully urges that a table such as that set forth within *Li, et al.*, at column 8, lines 42 *et seq.*, cannot be said to show or suggest a "text string data structure" as a "text string data structure" is a specific term of art within the computer area and cannot be modified by the Examiner for purposes of applying a graphic representation of a transform database as disclosed within *Li, et al.*

The Examiner also cites *Renegar*, U.S. Patent No. 6,024,571 in combination with *Li, et al.*, for a rejection of Claims 4-11, 14-18 and 21-25 under 35 U.S.C. § 103(a) and that rejection is also respectfully traversed. *Renegar* discloses a printed translation assistance learning aid which includes printed indicia including words and phrases in a first and second language and a phonetic pronunciation of those words. Nothing within *Renegar* shows or suggests in any way a "text string data structure" which is embedded within a computer usable medium as set forth within the present claims and no combination of *Renegar* with *Li, et al.*, can be said to show or suggest this novel data structure.

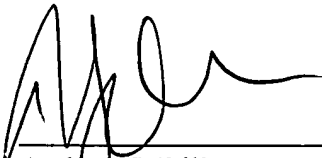
Further, as set forth expressly within Claims 10 and 16, the third character string within the text string data structure is expressly recited as being prefixed by "at least one character with a low sort value." The Examiner has referred expressly to these claims and directs the Applicant's attention to the rejection of Claim 6; however, the Applicant directs the Examiner's attention to the fact that the rejection of Claim 6, and the rejection of Claims 10 and 16, as well as *Li, et al.*, and *Renegar* are completely silent with respect to any suggestion of the prefixing of a third character string by "at least one character with a low sort value" and the Examiner is urged to consider this rejection is not well founded.

Applicant urges the Examiner to consider that data structures have been repeatedly found to be structure which must be considered in determining the patentability of an invention. *See In Re Lowry*, 32 U.S. P.Q.2d 1031 (Fed. Cir. 1994), wherein Applicant's invention concerned a data structure for storing, using and managing data in a computer memory. Similarly, *In Re Warmerdam*, 31 U.S. P.Q.2d 1754 (Fed. Cir. 1994), also addresses the subject of data structures. Applicant respectfully urges that the "text string data structure" of the present invention is neither shown nor suggested by *Li, et al.* or *Renegar*, whether considered alone or in combination as no combination of these references shows or suggests a text string data structure having multiple fields for representing a word such that the text string data structure may be manipulated within a data processing system and, by selecting one field or the other, the user may operate in one of two selected languages. Consequently, Applicant urges that Claims 1-25 define patentable subject matter over this combination of references and withdrawal of the Examiner's rejection and passage of this Application to issue is therefore respectfully requested.

No fee is believed to be required; however, in the event any additional fees are required, please charge IBM Corporation Deposit Account No. 09-0447. No extension of time is believed to be required; however, in the event any extension of time is required, please consider that

extension requested and please charge any associated fee and any additional required fees, to
IBM Corporation Deposit Account No. 09-0447.

Respectfully submitted,

A handwritten signature in black ink, appearing to read 'AJD', is written over a horizontal line.

Andrew J. Dillon

Reg. No. 29,634

BRACEWELL & PATTERSON, L.L.P.

Suite 350, Lakewood on the Park

7600B North Capital of Texas Highway

Austin, Texas 78731

(512) 542-2100

ATTORNEY FOR APPLICANTS

IN THE SPECIFICATION:

Please substitute the following paragraph for the "RELATED APPLICATIONS".

In present Specification at Page 1, lines 7-Page 2, line 6 as follows:

--The present invention is related to the subject matter of the following commonly
5 assigned, copending United States patent applications: serial no. [08]09/211,810 (Docket No. AT9-98-108) entitled "METHOD, SYSTEM, AND COMPUTER PROGRAM PRODUCT FOR PROVIDING A USER INTERFACE WITH ALTERNATIVE DISPLAY LANGUAGE CHOICES" and filed December 15, 1998; serial no. [08]09/211,809 (Docket No. AT9-98-158) entitled "METHOD, SYSTEM AND COMPUTER PROGRAM PRODUCT FOR CAPTURING
10 LANGUAGE TRANSLATION AND SORTING INFORMATION INTO A TEXT STRING CLASS" and filed December 15, 1998; serial no. [08]09/211,808 (Docket No. AT9-98-159) entitled "METHOD, SYSTEM AND COMPUTER PROGRAM PRODUCT FOR SORTING TEXT STRINGS" and filed December 15, 1998; serial no. [08]09/211,799 (Docket No. AT9-98-408) entitled "METHOD, SYSTEM AND COMPUTER PROGRAM PRODUCT FOR
15 ROTATING THROUGH A SEQUENCE OF DISPLAY STATES IN A MULTI-FIELD TEXT CLASS IN A GRAPHICAL USER INTERFACE" and filed December 15, 1998; serial no. [08]09/211,802 (Docket No. AT9-98-409) entitled "METHOD, SYSTEM AND COMPUTER PROGRAM PRODUCT FOR CONTROLLING THE GRAPHICAL DISPLAY OF MULTI-FIELD TEXT STRING OBJECTS" and filed December 15, 1998; serial no. [08]09/211,813
20 (Docket No. AT9-98-410) entitled "METHOD, SYSTEM AND COMPUTER PROGRAM PRODUCT FOR DISPLAYING THE CONTENTS OF ALL FIELDS IN A MULTI-FIELD TEXT STRING OBJECT" and filed December 15, 1998; serial no. [08]09/211,801 (Docket No. AT9-98-411) entitled "METHOD, SYSTEM AND COMPUTER PROGRAM PRODUCT FOR DYNAMIC LANGUAGE SWITCHING IN A MULTI-FIELD TEXT STRING OBJECT VIA
25 MESSAGING" and filed December 15, 1998; and serial no. [08]09/211,812 (Docket No. AT9-

98-578) entitled "METHOD, SYSTEM AND COMPUTER PROGRAM PRODUCT FOR AUTOMATIC CHARACTER TRANSLITERATION IN A TEXT STRING OBJECT" and filed December 15, 1998. The content of the above-referenced applications is incorporated herein by reference.--

streaming tape \strē'mēng tāp\ *n.* See tape (definition 1).

stream-oriented file \strēm'ōr'ē-ent-əd fīl\ *n.* A file used to store a fairly continuous series of bits, bytes, or other small, structurally uniform units.

street price \strēt' prīs\ *n.* The actual retail or mail-order price of a consumer hardware or software product. In most cases, the street price is somewhat lower than the "suggested retail price."

stress test \stres' test\ *n.* A test of a software or hardware system's functional limits, performed by subjecting the system to extreme conditions, such as peak volumes of data or extremes in temperature.

strikethrough \stri'k thrō\ *n.* One or more lines drawn through a selected range of text, usually to show deletion or the intent to delete, as in **strikethrough**.

string \strēng\ *n.* A data structure composed of a sequence of characters usually representing human-readable text.

string variable \strēng' vār'ē-ə-bl\ *n.* An arbitrary name assigned by the programmer to a string of alphanumeric characters and used to reference that entire string. See also string.

stroke \strōb\ *n.* A timing signal that initiates and coordinates the passage of data, typically through an input/output (I/O) device interface, such as a keyboard or printer.

stroke \strōk\ *n.* 1. In data entry, a keystroke—a signal to the computer that a key has been pressed. 2. In typography, a line representing part of a letter. 3. In paint programs, a "swipe" of the brush made with the mouse or keyboard in creating a graphic. 4. In display technology, a line created as a vector (a path between two coordinates) on a vector graphics display (as opposed to a line of pixels drawn dot by dot on a raster graphics display).

stroke font \strōk' font\ *n.* A font printed by drawing a combination of lines rather than by filling a shape, as with an outline font. Compare outline font.

stroke weight \strōk' wāt\ *n.* The width, or thickness, of the lines (strokes) that make up a character. See also font.

stroke writer \strōk' rī tər\ *n.* In video, a display unit that draws characters and graphic images as sets of strokes—lines or curves connecting

points—rather than as sets of dots, as on a typical raster-scan monitor. See also vector graphics.

strong typing \strong' tī'pēng\ *n.* A characteristic of a programming language that does not allow the program to change the data type of a variable during program execution. See also data type, variable. Compare weak typing.

structure \struk'chur\ *n.* 1. The design and composition of a program, including program flow, hierarchy, and modularity. 2. A collection of data elements. See also data structure.

structured graphics \struk'churd graf'iks\ *n.* See object-oriented graphics.

structured programming \struk'churd prō'gram-ēng\ *n.* Programming that produces programs with clean flow, clear design, and a degree of modularity or hierarchical structure. See also modular programming, object-oriented programming. Compare spaghetti code.

structured query language \struk'churd kwēr'ē lang-wəj, kwār'ē\ *n.* A database sublanguage used in querying, updating, and managing relational databases—the de facto standard for database products. Acronym: SQL (sē'kwəl, S'Q-L).

structured walkthrough \struk'churd wāk'thrō\ *n.* 1. A meeting of programmers working on different aspects of a software development project, in which the programmers attempt to coordinate the various segments of the overall project. The goals, requirements, and components of the project are systematically reviewed in order to minimize the error rate of the software under development. 2. A method for examining a computer system, including its design and implementation, in a systematic fashion.

STT \S'T-T'\ *n.* See Secure Transaction Technology.

stub \stub\ *n.* A routine that contains no executable code and that generally consists of comments describing what will eventually be there; it is used as a placeholder for a routine to be written later. Also called dummy routine. See also top-down programming.

StuffIt \stuf'it\ *n.* A file compression program originally written for the Apple Macintosh, used for storing a file on one or more disks. Originally shareware, StuffIt is now a commercial product for Macs and PCs that supports multiple compression techniques and allows file viewing. StuffIt files can